UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

TRAFFIC ANSI

AN EQUILIBRIUM TRAFFIC ASSIGNMENT PROGRAM

(ANSI FORTRAN version)

by

Sang Nguyen and Linda James-Lefebvre

ANSI version by André Babin

Ce rapport a été publié grâce à une subvention du
Fonds F.C.A.C. pour l'aide et le soutien à la recherche.

This is a revised version of Publication #17

Université de Montréal

Centre de recherche sur les transports - Publication #220

Août 1981 - révisé en avril 1984

Dépôt légal - 2e trimestre 1984

RÉSUMÉ

Le logiciel TRAFFIC effectue une allocation de la demande entre différentes paires origine-destination sur les arcs d'un réseau routier, de manière à minimiser le coût de parcours individuel de chaque utilisateur du réseau. Le coût de parcours de chaque arc est une fonction croissante du flot total empruntant l'arc, et aucune limite n'est imposée sur le flot total de chaque arc.

- - - - - - - - - - - - - - -

Purpose

To perform the assignment of traffic to links in a given transportation network to satisfy the demand between the origins and destinations in the network such that the cost to each individual user of the network is minimized. It is assumed that the travel cost on each link is an increasing function of the total link flow and no explicit capacity constraint is imposed on the network links.

General description

The program reads in the characteristics of the links of the network and the demands between origin/destination pairs of nodes (centroids). The algorithm used to perform the assignment is an iterative method developed by S. Nguyen* (algorithm A1 in reference). The user supplies parameters which determine the number of iterations which will be performed before the program stops and a report on the resulting solution is put out.

The program also allows the user to restart computation from an existing solution for the same network, and carry out further iterations as desired.

---

* "A Mathematical Programming Approach to Equilibrium Methods of Traffic Assignment with fixed demands" by S. Nguyen (Université de Montréal, Département d'informatique et de recherche opérationnelle, Publication #138).

CONTENTS

Purpose ii

General description ii

Contents. iii

General operating instructions. 2

Comments on program files 3

Input files 3

1. Program parameter and network description file 4

2. Origin/destination matrix file 6

3. Delay function coefficients file 7

4. Intermediate solution file 8

Work files. 8

Output files. 9

1. Cycle log. 9

2. Terminal solution report 10

3. SOLFIL Intermediate solution file. 10

Program and algorithm parameters. 10

Program structure 11

Use of DOUBLE PRECISION 12

Error messages. 14

Sample problem. 17

Corrections and modifications in this version 25

APPENDIX I - Source program listing 26

APPENDIX II - Sample problem file. 45

APPENDIX III - Sample problem output report 47

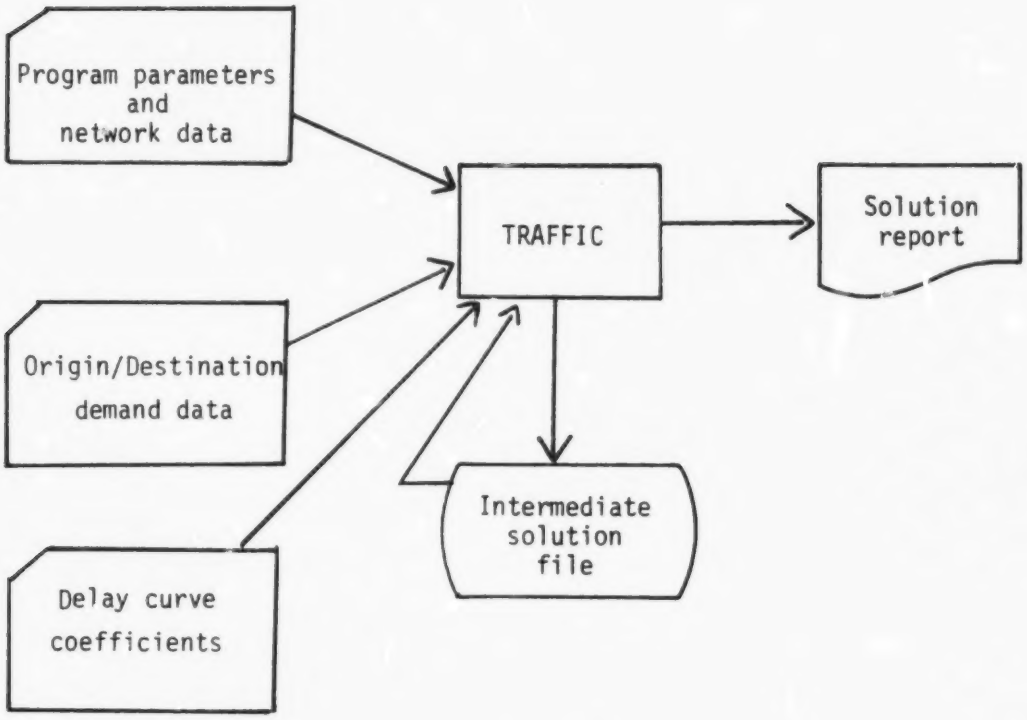

Figure 1 : Diagram showing Input and Output files

General operating instructions

The user must insert the code for the delay curves applicable for his network. (See the description of the Sample Problem for a detailed example.) This implies the following :

a) Main program TRAFFIC
   Insert statements to calculate the constant part of the objective function. (If this is the case).
   Insert statements in the vehicle hours and miles calculation to exclude dummy links, if any.

b) Function USERCT
   Insert the statements to calculate USERCT for link J.

c) Function SUMTRV
   Insert the statement to calculate the value of the variable part of the objective function.

d) Subroutine FINDCC
   Insert the statements to get the delay function coefficients for use in the function USERCT.

The user must also provide a SUBROUTINE SECOND (T) which returns the current time (seconds) as a real variable T.

For a system other than CDC, the first statement PROGRAM must be changed to a statement appropriate for the start of a main program on the user's system.

The program is then compiled and executed as for a standard Fortran program. Input data files, work files and output files are described below.

The program can be used in two ways (specified in the program parameters in an input file), as mentioned before.

a) to start from the demand between origin/destination pairs (initial run)
b) to start from an intermediate solution (restart).

Comments on program files

The input/output statements for all the files use integer variables to specify the unit numbers.

The files are currently numbered as follows :

1 ODFIL - Internal origin/destination matrix work file
5 Card input files (3)
- Program parameter and network description file
- Origin/destination matrix file
- Delay function coefficient file
6 Report file
7 SOLFIL - Intermediate solution file
8 NETFIL - Internal network description file

N.B. The Université de Montréal system control cards (using a CDC Cyber 173) include

LIB(C,TAPE7,N=TS01)

which causes the intermediate solution file to be saved (with name TS01)

Files 1 and 8 are internal work files which are automatically assigned space by the CYBER 173 system. On an IBM system, specific DD Statements would be needed to allocate space for these files as well as file 7.

Input files

The program always requires three input files

- Program parameter and network description file
- Origin/destination matrix file
- Delay function coefficient file

In addition for a restart run it requires

- Intermediate solution file (SOLFIL).

The Program parameter and network description file contains:

- report titles
- parameters to control the algorithm
- description of the network (links).

The origin/destination matrix file contains the demand for all the destinations for each origin. The delay curve coefficient file contains the four coefficients for each delay curve. The intermediate solution file contains the volumes and travel times for each link from the final solution of a previous computer run.

Descriptions of these files follow.

1. Program parameter and network description file (always required)

This file contains, in the following order:

- Report title cards (2)
- Comment card
- Parameter card
- Network size card
- Network data cards
- Blank card

The card contents and formats (in Fortran FORMAT notation) are:

a) Title cards: Contain the two lines used as a title on the output reports.

Card 1: cc 1-5 'TIT1 '
cc 6-45 Text for first line of title

Card 2: cc 1-5 'TIT2 '
cc 6-65 Text for second line of title

b) Comment card: Comment which is saved on the intermediate solution file and printed out when this file is used in a restart run.

Card 1: cc 2-80 Text of comment (cc 1 must be blank)

c) Parameter card: Algorithm and program parameters

Card 1:	cc 1-5		'PARM'
	cc 6-15	F10.0	Epsilon
	cc 16-20	F5.0	Maximum run time
	cc 21-25	I5	Maximum number of cycles
	cc 26-30	I5	Not used
	cc 31-35	I5	Postoptimize indicator
			= 0 : Initial run
			> 0 : Restart
	cc 36-40	I5	Not used
	cc 41-45	I5	Print control indicator
			= 0 : Suppress solution detail
			> 1 : Print solution detail

d) Network size card

Card 1 :	cc 1-5		'SIZE'
	cc 6-10	I5	Number of nodes (total including centroids)
	cc 11-15	I5	Number of links
	cc 16-20	I5	Number of centroids (origins and/or destinations).
	cc 21-25	I5	Number of delay functions

The number of links must correspond with the number of network data cards which follow.

The number of centroids must correspond with the number of sets on the origin/destination matrix file.

The number of delay functions must correspond with the number of delay function cards.

e) Network data cards: 1 card for each link

cc 1-5		Identification (not checked)
cc 6-9	I4	Link origin node number
cc 11-14	I4	Link destination node number
cc 16-19	I4	Link length (miles x 100)
cc 21-23	I3	Number of lanes x 10
cc 24-25	I2	Delay function number
cc 27-30	I4	Observed travel speed (mph x 10) } if available
cc 32-36	I5	Observed travel time (mins x 100) } if available
cc 38-42	I5	Observed volume
cc 44-46	I3	Link type
cc 48-50	I3	Not used
cc 52-54	I3	Not used

The network data cards must be sorted by the link origin node and must be followed by a blank line terminator.

Note that centroids must be numbered starting with 1 up to the number of centroids as specified on the Network size card, followed by the other nodes numbered consecutively.

2. Origin/destination matrix file (always required)

This file contains one set of data for each centroid, on one or more cards. The file must contain at least one card for each centroid even if the centroid is not an origin. The centroids need not appear in numerical order but the data cards for a given centroid must be together. The card contents and formats (in Fortran FORMAT notation) are

cc 1-5	I5	Centroid (node) number
cc 6-10	I5	Total number of destinations with non-zero demands originating at this centroid
cc 11-13	I3	First destination node number
cc 14-17	I4	Volume to the first destination
cc 18-20	I3	2nd destination node number
cc 21-24	I4	Volume to 2nd destination
etc.		
cc 74-76	I3	10th destination node number
cc 77-80	I4	Volume to 10th destination

Note: The set of data for a centroid may be on one or more cards. All the cards have the same format, with the centroid number and total (for the set) number of destinations repeated in the second and any following cards. The last card in a set may have data for less than 10 destinations.

The centroid data sets may appear in any order and the destination nodes may be in any order within a set.

3. Delay function coefficients file (always required)

This file contains 1 card for each delay function number even if the function is not used. Each card contains the delay function number and four coefficients. The cards must be sorted by delay function number.

The card contents and format (in FORTRAN format notation) are

cc 1- 5	I5	delay function number, I
cc 6-15	F10.5	coefficient ALPHA(I)
cc 16-25	F10.5	coefficient BETA(I)
cc 26-35	F10.5	coefficient GAMMA(I)
cc 36-45	F10.5	coefficient DELTA(I)

See the description of the sample problem code for an example of the use of these coefficients.

4. Intermediate solution file (required for a restart run only)

This file is created when the program terminates a run, when one of the three possible terminating conditions is satisfied and the solution is printed. When a restart run is made, the intermediate solution file must have been produced using the identical network but changes in characteristics for existing links are allowed.

The file has the following structure.

Record 1 : Format (10A8). Comments card from the run creating the file.

Records 2 to i : Format (10F8.0). Cumulated volume on each link, for the the solution at termination.

Records i+1 to j : Format (10F8.2). Travel time predicted for each link, for the solution at termination (not used for restart).

Records j+1 to k : Format (20I4). Contents of internal vector KD which gives the corresponding centroid number for each set on the origin/destination file. (Not used for restart).

where i = Number of links/10 + 1

j = Number of links/10 + i + 1

k = Number of centroids/20 + j + 1

Work files

The program requires the following work files:

- ODFIL (File 1) Origin/destination demand data
- NETFIL (File 8) Network data file.

1. ODFIL: Origin/destination demand data

This file is an unformatted Fortran file containing one record for each origin centroid; that is each centroid with a non-zero demand. Each record

consists of the vector ODIJ (150) which specifies the demand to each destination (maximum 150) from that origin.

2. NETFIL: Network data file

This file is a formatted Fortran file containing one record for each link in the network in exactly the same format as the input Network data cards, previously described. It is used to produce the detailed solution report.

Output files

The program produces a log, giving the status at the end of each cycle in the algorithm. It also produces the following files when one of the terminating conditions is satisfied.

- Cycle log
- Terminal solution report
- SOLFIL : Intermediate solution file.

1. Cycle Log

The log shows

- The cycle number
- Value of the objective function for the current solution
- Gap between the lower bound and the current value of the objective function, as a percentage of the objective function value
- Value of the step length, (λ)
- cumulative run time (seconds)

2. Terminal solution report

This report gives

- the condition which terminated the iterative procedure
- the value of the terminating condition (Epsilon, maximum time, maximum cycles)
- the time taken in execution of various sections of the algorithm (run statistics)
- the total vehicle hours, total vehicle miles and average speed
- a report containing one line for each link giving the network input data plus the predicted travel time and volume for the link. (Can be suppressed by the print control parameter. An example of this report is contained in Appendix III).
- a report containing one line for each delay function number giving the function coefficients (can be suppressed by the print control parameter. An example of this report is contained in Appendix III).

3. SOLFIL : Intermediate solution file

This file contains the cumulated volumes and travel times for the solution at termination. The format is described in the section on "Input files".

Program and algorithm parameters

Program and algorithm parameters are specified on the Parameter card in the "Program parameter and network description" file. The parameters are used as follows:

1. Epsilon

The iterative procedure terminates and the solution is printed if

Change in objective function between last 2 cycles $\leq$ Epsilon.

2. Maximum run time

The iterative procedure terminates at the end of the cycle and the solution is printed when

Total run time $\geq$ Maximum run time.

3. Maximum number of cycles

The iterative procedure terminates and the solution is printed at the end of the cycle when

Cycle number = number of cycles.

4. Postoptimize indicator

This parameter determines if the program will construct the initial (all or nothing) solution as a starting solution or will use an intermediate solution from SOLFIL.

Indicator = 0	Construct initial solution
Indicator > 0	Use intermediate solution.

5. Print control indicator

This parameter controls the printing of the detailed solution report (for each link in the network) and the delay function report.

Indicator = 0	Report are suppressed
Indicator = 1	Report are printed

The objective function at each iteration and the run statistics are always printed.

Program structure

The program consists of a main program, subroutines and functions, all written in ANSI STANDARD FORTRAN IV as follows :

Main program :	TRAFFIC
Subroutines :	NETWRK
	LOADTR
	MINTRE
	GSM
	FINDCC
Functions :	SUMTRV
	USERCT

The subroutine/function hierarchy is illustrated in Figure 2. Note that the functions are used in the main program and its subroutines.

Use of DOUBLE PRECISION

On a computer with a word length of 60 bits such as the CDC Cyber series, TRAFFIC will run more efficiently (faster) if the following changes are made :

- Replace the DOUBLE PRECISION statement by REAL statements in the main program and subroutines SUMTRV (2 statements), GSM
- Replace function DABS by function ABS in the main program.

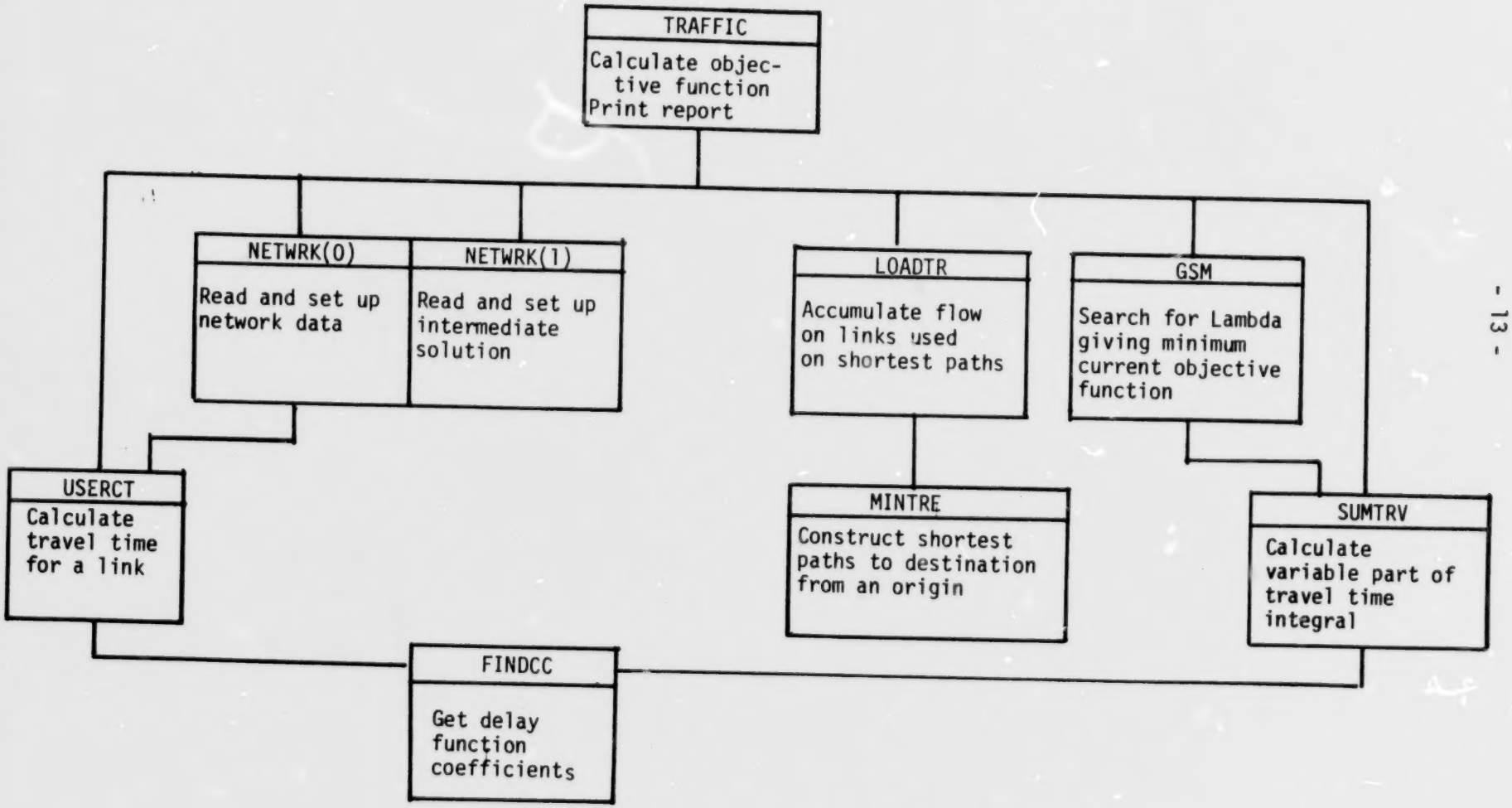

Figure 2 : Main program, subroutine and function hierarchy

Error messages

Messages produced by the program have the following format:

XNN Text

where X = I for "Information" messages (no error)
= E for "Error" messages
NN = 2 digit number of the message
Text = Message text.

The message and explanations follow in sequence according to NN.

Message I01 RUN TERMINATED ... TIME LIMIT : t

Explanation Run time ≥ t = Maximum run time parameter.

Message I02 RUN TERMINATED ... CYCLE LIMIT : m

Explanation Last cycle number = m = Maximum number of cycles parameter.

Message E03 LINKS NOT SORTED. LINE : n

Explanation The n th network data card contains a link origin node number lower than that of the previous network data card.

Action Sort the network data cards by origin node.

Message E04 NO. CARDS .NE. NLINKS n1 n2

The number of network data cards n1 is not equal to the number of links n2 specified in the Network size card.

Action Correct the data.

Message E05 ERROR IN INPUT OD MATRIX. TYPE : t

Explanation (t= 1) : The origin node number on an origin/destination demand card is either zero or the origin set has already been completed and this card is in excess of the number of destinations for that origin.

Action (t = 1) : Correct the number of destinations

Explanation (t = 2) : End-of-file has been encountered on the origin/destination matrix file before sets of origin/destination demand for all centroids have been found.

Action (t = 2) : Correct the origin/destination matrix file (see page 6).

Message E06 NSET IN MINTRE OVERFLOW. CURRENT ORIGIN IS n

Explanation The capacity of the heap structure (NSET) has been exceeded during processing for origin n.

Message I07 PARTIAL SOLUTION INPUT i

Explanation The postoptimal solution parameter was i and the intermediate solution file will be used to set up the initial solution.

Message I09 INITIAL SOLUTION - VALUE OF OBJ. FUNCTION f

Explanation The initial solution (all or nothing or intermediate) has been set up. The objective function value is f.

Message I10 DEL(F) < EPSILON e

Explanation The run was terminated since, between the last 2 cycles, Change in the objective function (DEL(F)) ≤ e.

Message E11 TITLE CARDS MISSING aaaaa bbbbb

Explanation The title cards which should be at the beginning of the "Program parameter and network description" file have not been found. aaaaa and bbbbb are the contents of cc 1-5 of the 2 records. (These should be TIT1 and TIT2 respectively for title cards).

Action Correct the data.

Message E12 PARAMETER CARD MISSING aaaaa

Explanation The parameter card (fourth card in the "Program parameter and network description" file) has not been found. aaaaa are the contents of cc 1-5 of the fourth card which should be PARM.

Action Correct the data.

Message E13 SIZE CARD MISSING aaaaa

Explanation The size card (fifth card in the "Program parameter and network description "file) has not been found. aaaaa are the contents of cc 1-5 of the fifth card which should be SIZE.

Action Correct the data.

Message E20 TOO MANY INPUT ERRORS n

Explanation Self explanatory. n = number of errors found. The program terminates

Action Correct the data errors indicated in preceding messages.

Message E21 ERROR IN DELAY FUNCTION DATA, TYPE : t

Explanation (t=1) Number of delay functions specified on the SIZE card is not valid.

Action (t=1) Correct the SIZE card

Explanation (t=2) A delay function number found in the network data is higher than the number of functions specified in the SIZE card

Action (t=2) Correct the data or the SIZE card and coefficient file.

Message	E22 DELAY FUNCTION COEFFICIENTS NOT SORTED. LINE : n
Explanation	The n delay function coefficient card contains a delay function number that is lower than the preceding card.
Action	Sort the delay function coefficient file by delay function number. There must be one card for each delay function up to the maximum specified in the SIZE card.

Sample problem

The problem used as a sample problem for the program TRAFFIC is based on a test problem proposed by A. Gibert and M. Sakarovitch* in 1968.

The network is shown in Figure 3. It has 17 nodes of which 4 are centroids. In this simple problem, the centroids are either origins (2) or destination (2) but this would not necessarily be true of other problems. The data describing the links are shown in Figure 4. The flow demand between the origins and destinations (OD matrix) is shown in Figure 5. The delay function coefficients are shown in Figure 6.

In this formulation the nodes 1, 2, 3 and 4 have been added to illustrate the dummy links which carry the total flow into (or from) a centroid. For example the link from node 3 to 13 is a dummy link which carries the aggregate flow into node 13. The dummy links are identified by a unique delay function number (1) and they have zero length.

---

* Institut de Recherche des Transports, Paris, France.

The delay function for this problem is the same for all links and is the BPR function, viz.

$$t = \frac{\ell}{\gamma} \left(1 + \alpha\left(\frac{v}{C}\right)^{\beta}\right)$$

where t is the user cost or travel time for a link

ℓ is the length of the link

v is the volume on the link

α is a constant coefficient (= 0.02)

β is a constant coefficient (= 5.0)

γ is free flow speed

C is capacity of the link = δ x number of lanes

δ capacity per lane or the link

The objective function is equal (in all cases) to the sum over all links of the integral of the delay function for each link with respect to volume, integrated from zero to the volume currently assigned to the link. This is broken down into two parts, one variable, the other constant.

For the sample problem, the constant part is zero and the variable part for each link is

$$\frac{\ell}{\gamma} v \left(1 + \frac{\alpha}{\beta+1} \left(\frac{v}{C}\right)^{\beta}\right)$$ (Summed for all links by function SUMTRV)

using the same notation as above.

In order to use TRAFFIC to solve the problem, the user must prepare a description of the network and OD matrix as described in the section on Input files (Program parameter and network description file and Origin/Destination matrix file). He must also insert his own code for the delay functions into

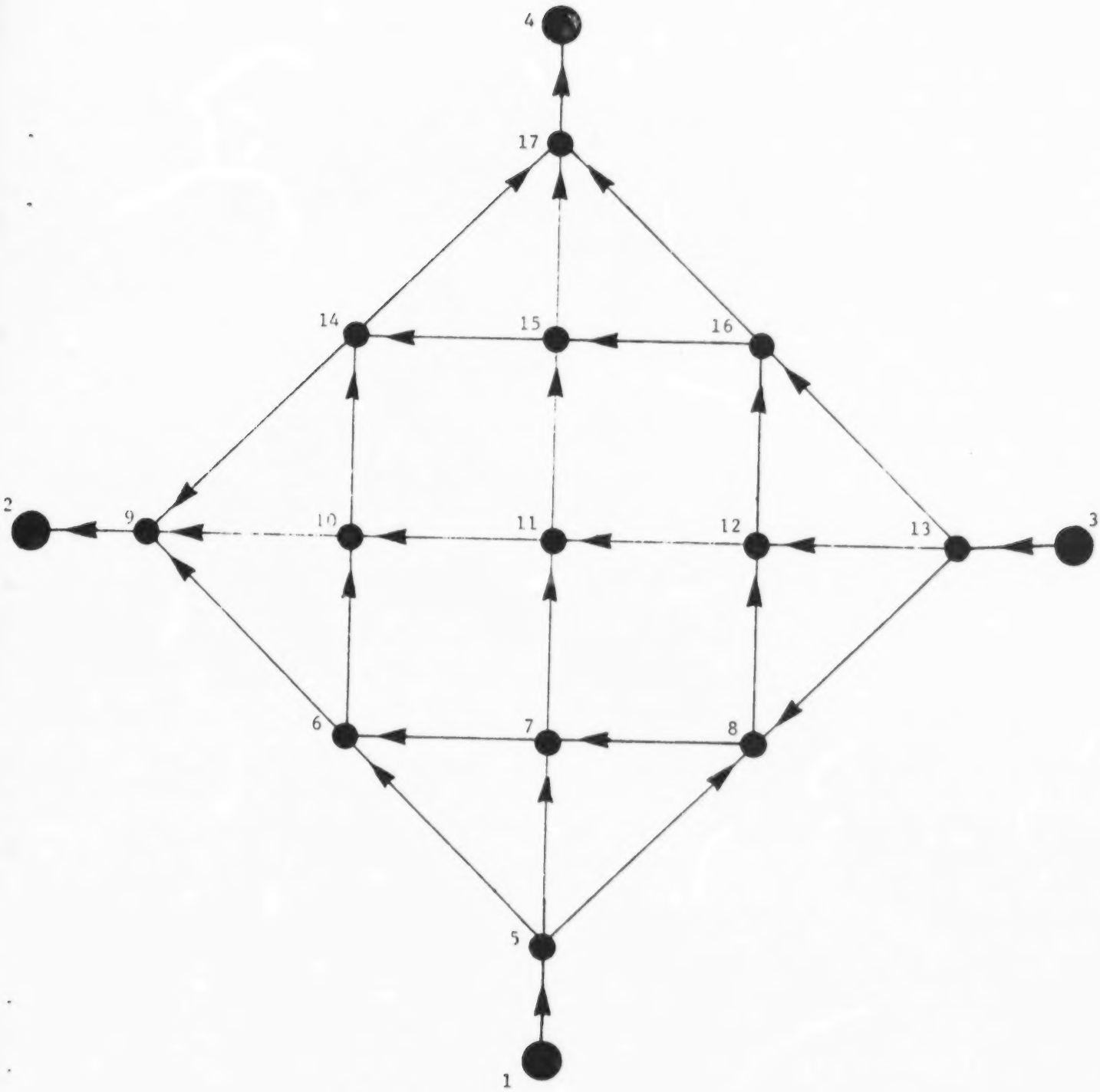

Figure 3 : Sample problem network

Origin node	Destination node	Length	Number of lanes x 10	Delay function
1	5	0	10	1
3	13	0	10	1
5	6	300	10	2
5	7	200	10	2
5	8	100	10	2
6	9	2300	10	2
6	10	1700	10	2
7	6	400	10	2
7	11	1200	10	2
8	7	500	10	2
8	12	600	10	2
9	2	0	10	1
10	9	2100	10	2
10	14	1000	10	2
11	10	1600	10	2
11	15	700	10	2
12	11	1500	10	2
12	16	1100	10	2
13	8	1800	10	2
13	12	1900	10	2
13	16	2200	10	2
14	9	2000	10	2
14	17	900	10	2
15	14	800	10	2
15	17	1400	10	2
16	15	2400	10	2
16	17	1300	10	2
17	4	0	10	1

Figure 4 : Network link description

Origin node	Destination node	Demand
1	2	2500
1	4	2500
3	2	2500
3	4	2500

Figure 5 : Flow demand (OD matrix)

Delay function	Coefficient α	Coefficient β	Coefficient γ Free flow speed	Coefficient δ Capacity per lane
1	0.00	0.0	60.	0.
2	0.02	5.0	30.	1000.

Figure 6 : Delay function coefficient table

---

the main program and functions USERCT and SUMTRV as described below.

A listing of the input for the sample problem is contained in Appendix II. The output report for the sample problem is contained in Appendix III. Appendix I contains the source program listing, including the sample problem user code.

1. Program parameter and network description file

Refer to Figure 7.

The PARM card contains parameters as follows :

Epsilon = .0001
Maximum run time = 100 secs.
Maximum number of cycles = 20
Postoptimize indicator = 0 (Initial run)
Print control indicator = 1 (print solution detail)

The SIZE card contains :

Number of nodes = 17
Number of links = 20
Number of centroids = 4
Number of delay functions = 2

(Note : Centroids are numbered from 1 to the number of centroids)

Program parameter and network description file

```
TIT1 CONGESTED ASSIGNMENT TEST
TIT2 PROBLEM GS
 SAMPLE PROBLEM FOR ALGORITHM A1 (FEB 12 1975)
PARM       .0001    20    20          0          1
SIZE     17    28     4     2
GS74      1     5     0    10 1
GS74      3    13     0    10 1
GS74      5     6   300    10 2
GS74      5     7   200    10 2
GS74      5     8   100    10 2
GS74      6     9  2300    10 2
GS74      6    10  1700    10 2
GS74      7     6   400    10 2
GS74      7    11  1200    10 2
GS74      8     7   500    10 2
GS74      8    12   600    10 2
GS74      9     2     0    10 1
GS74     10     9  2100    10 2
GS74     10    14  1000    10 2
GS74     11    10  1600    10 2
GS74     11    15   700    10 2
GS74     12    11  1500    10 2
GS74     12    16  1100    10 2
GS74     13     8  1800    10 2
GS74     13    12  1900    10 2
GS74     13    16  2200    10 2
GS74     14     9  2000    10 2
GS74     14    17   900    10 2
GS74     15    14   800    10 2
GS74     15    17  1400    10 2
GS74     16    15  2400    10 2
GS74     16    17  1300    10 2
GS74     17     4     0    10 1
```

Origin destination matrix file

```
1    2  42500   22500
2    0
3    2  22500   42500
4    0
```

Delay function coefficient file

```
1        0.0         0.0        60.         0.0
2       0.02         5.0        30.      1000.0
```

Figure 7 : Sample problem input files

2. Origin/destination matrix file

Refer to Figure 7.

The first card specifies :

Origin node = 1
Number of destinations = 2
1st destination node = 4
Volume of flow from the origin to the 1st destination = 2500
2nd destination node = 2
Volume of flow from the origin to the 2nd destination = 2500

The second card specifies that there are no destinations from centroid 2 i.e. it is not an origin.

3. Delay function coefficient file

The second card specifies :

Delay function number = 2
Coefficient α = .02
Coefficient β = 5.0
Coefficient γ = 30. (free flow speed)
Coefficient δ = 1000. (capacity per lane)

User supplied code (Sample problem)

The user supplied code for the sample problem can be found in the program listing, in the main program and the functions USERCT and SUMTRV. In each case it is located between two comments cards containing all plus signs (C+++++ ....). Comments and clarification on this code follow.

1. Main program

a) Calculation of constant part of objective function.

The user supplied code sets CONST to zero.

b) Omit dummy links in calculation of vehicle hours and miles.

The user supplied code tests the delay function number. If it is equal to one (dummy links) the calculations of vehicle hours and vehicle miles are bypassed for that link.

2. Function SUMTRV

This function calculates the variable part of the objective function as described on page 18.

The user supplied code increments SUM for each link J with delay function JC within the loop DO 100 I = 1, NLINK.

3. Function USERCT(J)

This function calculates the "user cost" i.e. the travel time for link J with is the value of the delay function for link J as described on page 18.

The user supplied code calculates the value of the delay function in USERCT

4. Subroutine FINDCC

This subroutine gets the delay function coefficients for the delay function LTYPE for use in USERCT, that is

SPDFF	free flow speed (γ in the sample problem)
CAP	capacity per lane (δ in the sample problem)
A	α (BPR coefficient α)
B	β (BPR coefficient β)

α, β, γ and δ are read in from the input file and stored in vectors ALPHA, BETA, GAMMA and DELTA.

ALPHA(I), BETA(I) etc. contain the value of α, β, etc., for delay function number I.

Corrections and modifications in this version

This version of TRAFFIC corrects the following errors in the original TRAFFIC program :

1) Error in the sample problem code in subroutines SUMTRV and USERCT.
   NULA contains the number of lanes x 10 but was used as if it was x 100.
   This has been corrected in the new sample problem code in these routines.
2) Error in the documentation of the network data cards.
   The link length in CC 16-19 should be the length specified as miles x 100.

This version of TRAFFIC contains the following modifications :

1) The program conforms to ANSI Fortran IV standards.
2) The delay function coefficients are read from the input file. Four coefficients are allowed for each function number.
3) The delay functions in the sample problem code are BPR functions instead of exponential functions.
4) The cycle log output has been improved.

APPENDIX I

SOURCE PROGRAM LISTING

```
      PROGRAM TRAFFIC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE8,
     +                TAPE7)
C
C--------THIS PROGRAM PERFORMS THE ASSIGNMENT OF TRAFFIC TO LINKS IN A
C        GIVEN TRANSPORTATION NETWORK TO SATISFY THE DEMAND BETWEEN THE
C        ORIGIN/DESTINATION PAIRS IN THE NETWORK SUCH THAT THE COST TO
C        EACH INDIVIDUAL USER F THE NETWORK IS MINIMIZED.
C
C------ THE PROGRAM IS BASED ON THE ALGORITHM DESCRIBED IN  " A
C       MATHEMATICAL PROGRAMMING APPROACH TO EQUILIBRIUM METHODS
C       OF TRAFFIC ASSIGNMENT WITH FIXED DEMANDS"  BY
C       SANG NGUYEN (UNIVERSITE DE MONTREAL, DEPT. DE
C       INFORMATIQUE, TECHNICAL REPORT, NUMBER 138, AUGUST 1973).
C
C       THIS PROGRAM IS THE ANSI STANDARD FORTRAN IV
C       VERSION OF THE PROGRAM DESCRIBED IN THE PUBLICATION:
C        "TRAFFIC AN EQUILIBIRUM TRAFFIC ASSIGNMENT PROGRAM"
C        BY SANG NGUYEN AND LINDA JAMES
C        C.R.T. UNIVERSITY OF MONTREAL, PUBL. # 17,
C       AND WAS CODED BY ANDRE BABIN.
C       THE CORRESPONDING PUBLICATION IS:
C         "TRAFFIC AN EQUILIBRIUM TRAFFIC ASSIGNMENT PROGRAM"
C         (ANSI FORTRAN IV  VERSION)
C       BY SANG NGUYEN AND LINDA JAMES-LEFEBVRE.
C       ANSI VERSION BY ANDRE BABIN.
C         CENTRE DE RECHERCHE SUR LES TRANSPORTS
C         UNIVERSITY OF MONTREAL,  PUBL. # 220
C
C       NOTE: SUBROUTINE  SECOND  IS NOT ANSI STANDARD.
C             IT RETURNS THE CURRENT TIME IN SECONDS.
C
      COMMON /NET1/ NNODE,NLINK,NCENT,NACTIV,KD(150),NOLAB(1300),
     +              NO(3000),ND(3000),NULACO(3000),DIST(3000)
      COMMON /NET2/ VCUM(3000),TRAV(3000), FLOW(3000),ODIJ(150)
      COMMON        WKMEM(4300)
      COMMON /PARAM/ EPSI , TIMEMA, MAXCYC, MAXIT, IPCH, IPRT,IPOST
      COMMON /STAT/ RUNSEC(10), ITERAT(10)
      COMMON /COEF/ ALPHA(20),BETA(20),GAMMA(20),DELTA(20),NDATA
      COMMON /EDIT/ LINE(11),L(26),K(26),TITLE(50)
      COMMON /UNIT/ CARD,ODFIL,NETFIL,SOLFIL
      INTEGER CARD,ODFIL,NETFIL,SOLFIL,LLINE(40)
      DOUBLE PRECISION CONST,OBF,SUM,SUMTRV,LAMBDA,VAL,BOUND,GAP,ACCUR
      EQUIVALENCE (NORIG,LINE(1)),(NDEST,LINE(2)),(IDIST,LINE(3)),
     +            (NULA,LINE(4)),(DEFUNC,LINE(5)),(ISPEED,LINE(6)),
     +            (ITOBS,LINE(7)),(IVOBS,LINE(8)),(LTYPE,LINE(9))
      EQUIVALENCE (RDSEC,RUNSEC(1)),(TSLSEC,RUNSEC(2)),
     +            (PRTSEC,RUNSEC(3)),(GSMSEC,RUNSEC(4)),
     +            (PTHSEC,RUNSEC(5))
C
      DATA MXRUNS/5/
C
 1002 FORMAT(/,36H I01 RUN TERMINATED.....TIME LIMIT :,E15.7,//)
 1003 FORMAT(/,37H I02 RUN TERMINATED ... CYCLE LIMIT :,I8,/)
 1013 FORMAT(/,30H  OBJECTIVE FUNCTION TOLERANCE,5X,1H=,E15.7,/,
     +       18H  LAMBDA TOLERANCE,17X,1H=,E15.7,/,
     +       12H  TIME LIMIT,23X,1H=,E15.7,/,
     +       18H  NUMBER OF CYCLES,17X,1H=,I8)
 1014 FORMAT (/,22H TOTAL VEHICLE HOURS =,F12.2,/,
     +        /,22H TOTAL VEHICLE MILES =,F12.2,/,
     +        /,22H AVERAGE SPEED       =,F12.2,//)
 1019 FORMAT (/,35H  CPU SECS FOR - NETWORK DATA INPUT,13X,1H=,F10.2,/,
```

```
     +        16X,25H- SET UP INITIAL SOLUTION,7X,1H=,F10.2,/,                  640
     +        16X,31H- RE-ESTABLISH PARTIAL SOLUTION,1X,1H=,F10.2,/,            650
     +        16X,31H- GOLDEN SECTION METHOD (TOTAL),1X,1H=,F10.2,/,            660
65   +        16X,31H- FINDING SHORTEST PATH (TOTAL),1X,1H=,F10.2,//)           670
2000 FORMAT (1H1,100X,6HPAGE :,I4,//,30X,20A2,/,18X,30A2,//,                   680
     +        4X,46HLINK     NODE     NODE     LENGTH     NUMB    DELAY,       690
     +        4X,40HOBSERVED    OBSERVED    OBSERVED    LINK,                 700
     +        2X,20HPREDICTED  PREDICTED,/,                                    710
70   +        4X,19HIDEN     FROM      TO,14X,24HLANES   FUNCT      TRAVEL,    720
     +        6X,47HTRAVEL       VOLUME      TYPE    TRAVEL      VOLUME,/,     730
     +        55X,5HSPEED,8X,4HTIME,26X,4HTIME,/,                              740
     +        29X,5HMILES,21X,5H(MPH),7X,6H(MINS),24X,6H(MINS),//)             750
2003 FORMAT (1H1,30X,20A2,/,18X,30A2,//)                                       760
2005 FORMAT (4X,3(I4,4X),F6.2,4X,F3.1,5X,I2,8X,F4.1,6X,F6.2,4X,I8,             770
75   +        7X,I2,3X,F6.2,7X,I5)                                             780
2004 FORMAT(5X,3(I4,1X),I3,I2,1X,I4,1X,2(I5,1X),3(I3,1X))                      790
2006 FORMAT (4X,3(I4,4X),F6.2,4X,F3.1,5X,I2,8X,F4.1,6X,F6.2,4X,                800
     +        8H  NA     ,7X,I2,3X,F6.2,7X,I5)                                810
80   2011 FORMAT(10F8.0)                                                       820
2012 FORMAT (10F8.2)                                                           830
2013 FORMAT (20I4)                                                             840
2014 FORMAT(1H1,//,30X,27HDELAY FUNCTION COEFFICIENTS,//,                      850
     +        5X,21HDELAY FUNCTION NUMBER,5X,5HALPHA,10X,                      860
85   +        5H BETA,10X,5HGAMMA,10X,5HDELTA,//)                              870
2015 FORMAT(/,13X,I3,10X,4(F10.4,5X))                                          880
2050 FORMAT(//,16X,5HCYCLE,10X,9HOBJECTIVE,14X,3HGAP,9X,6HLAMBDA,              890
     +      8X,10HCUMULATIVE,/,16X,6HNUMBER,9X,8HFUNCTION,16X,1H%,            900
     +      24X,9HRUN  TIME,/,80X,9H(SECONDS),/)                              910
90   3000 FORMAT(16X,I4,8X,E16.10,6X,F7.1,                                     920
     +        7X,F7.4,9X,F7.2,/)                                               930
3003 FORMAT(44H1109 INITIAL SOLUTION-VALUE OF OBJ. FUNCTION,E20.10)            940
3004 FORMAT(2X,21HI10 DEL(F) < EPSILON ,E20.10)                                950
4001 FORMAT(40A2)                                                              960
95   C                                                                         970
     C  INITIALISATION                                                         980
     C                                                                         990
     C                                                                        1000
     C     TAPE FILES                                                         1010
100  C                                                                        1020
           CARD=5                                                            1030
           ODFIL=1                                                           1040
           NETFIL=8                                                          1050
           SOLFIL=7                                                          1060
105  C                                                                        1070
     C                                                                        1080
           ACCUR=0.01                                                        1090
     C                                                                        1100
     C     ACCUR IS REQUIRED PRECISION IN SEARCH FOR                          1110
110  C      LAMBDA  (SUBROUTINE GSM)                                          1120
     C                                                                        1130
           BOUND=-1000000000000.                                             1140
           CALL SECOND (TBEGIN)                                              1150
           DO 50 I=1,10                                                      1160
115        RUNSEC(I)=0.                                                      1170
        50 ITERAT(I)=0                                                       1180
     C--------READ NETWORK DATA                                              1190
           CALL NETWRK(0)                                                    1200
           CALL SECOND(TTOUT)                                                1210
120        RDSEC=TTOUT-TBEGIN                                                1220
           TTIN =TTOUT                                                       1230
     C--------CHECK FOR POST OPTIMIZATION                                    1240
```

```
      IF(IPOST.LT.1)GOTO 60                                                    1250
C                                                                              1260
125   C     ENTRY PARTSOL                                                      1270
C                                                                              1280
      CALL NETWRK(1)                                                           1290
          CALL SECOND(TTOUT)                                                   1300
          PRTSEC = TTOUT - TTIN                                                1310
130       GO TO 70                                                             1320
C                                                                              1330
C--------SET UP INITIAL (ALL OR NOTHING) SOLUTION                              1340
C                                                                              1350
   60 REWIND ODFIL                                                             1360
135   DO 40 I=1,NLINK                                                          1370
      VCUM(I)=0.0                                                              1380
      FLOW(I)=0.0                                                              1390
   40 TRAV(I)=USERCT(I)                                                        1400
      DO 42 N=1,NCENT                                                          1410
140       NN=KD(N)                                                             1420
          IF(NN.LE.0)GOTO 42                                                   1430
C--------READ DEMAND FOR THIS ORIGIN                                           1440
          READ (ODFIL) ODIJ                                                    1450
C--------ACCUMULATE ALL OR NOTHING SOLUTION (FLOW) FOR THIS ORIGIN              1460
145       CALL LOADTR(NN)                                                      1470
   42     CONTINUE                                                             1480
C                                                                              1490
      DO 43 I=1,NLINK                                                          1500
      VCUM(I)=FLOW(I)                                                          1510
150   TRAV(I)=USERCT(I)                                                        1520
   43 CONTINUE                                                                 1530
      CALL SECOND(TTOUT)                                                       1540
      TSLSEC=TTOUT-TTIN                                                        1550
C                                                                              1560
155   70 CONTINUE                                                              1570
      TTIN=TTOUT                                                               1580
      KOPT=0                                                                   1590
      NUMCYC=1                                                                 1600
C                                                                              1610
160   C--------CALCULATE OBJECTIVE FUNCTION FOR CURRENT SOLUTION                1620
C                                                                              1630
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++1640
C         SUPPLIED BY USER                                                     1650
C                                                                              1660
165   C         ** SAMPLE PROBLEM CODE **                                      1670
C                                                                              1680
      CONST=0.0                                                                1690
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++1700
C                                                                              1710
170   C     CONST = CONSTANT TERM OF OBJECTIVE FUNCTION.                       1720
C               = TOTAL COST, GIVEN USERCT VALUES CALCULATED FOR ZERO          1730
C                 VOLUMES                                                      1740
      OBF=SUMTRV (0.00)+CONST                                                  1750
C--------SUMTRV(0.0) = VARIABLE TERM OF OBJECTIVE FUNCTION                     1760
175   C                   = TOTAL COST, GIVEN USERCT VALUES CALCULATED         1770
C                           USING INITIAL SOLUTION VOLUMES.                    1780
C                                          (LAMBDA = 0.0)                      1790
      WRITE(6,3003) OBF                                                        1800
      WRITE(6,2050)                                                            1810
180   C                                                                        1820
  200  CONTINUE                                                                1830
      REWIND ODFIL                                                             1840
      DO 190 I=1,NLINK                                                         1850
```

```
  190     FLOW(I)=0.00                                                    1860
185     C                                                               1870
        C--------ACCUMULATE TRIAL FLOWS FOR NEW SOLUTION USING CURRENT  1880
        C         SOLUTION VOLUMES TO CALCULATE COST                    1890
        C                                                               1900
              DO 201 N=1,NCENT                                          1910
190           NN=KD(N)                                                  1920
              IF(NN.LE.0)GOTO 201                                       1930
        C--------READ DEMAND FOR THIS ORIGIN                           1940
              READ(ODFIL) ODIJ                                          1950
        C--------INCREMENT LINK FLOWS WITH THOSE FROM THIS ORIGIN       1960
195           CALL LOADTR(NN)                                           1970
          201 CONTINUE                                                  1980
        C                                                               1990
        C--------CALCULATE INSTANTANEOUS RATE OF CHANGE OF OBJECTIVE FUNCTION 2000
        C         ALONG THE DESCENT DIRECTION                           2010
200     C                                                               2020
              SUM=0.00                                                  2030
              DO 202 I=1,NLINK                                          2040
                 FLOW(I)=FLOW(I)- VCUM(I)                               2050
          202    SUM=SUM+ TRAV(I)*FLOW(I)                               2060
205     C                                                               2070
        C--------FLOW(I) IS NOW DESCENT DIRECTION FOR LINK I            2080
        C--------FIND LAMBDA TO MINIMIZE OBJECTIVE FUNCTION ALONG THE CURRENT 2090
        C         DESCENT DIRECTION.                                    2100
        C         VAL+CONST  = VALUE OF THE OBJECTIVE FUNCTION FOR THE  2110
210     C                      OPTIMUM LAMBDA.                          2120
              CALL SECOND(TTIN)                                         2130
              CALL GSM(LAMBDA,1.00,ACCUR,VAL)                           2140
              CALL SECOND(TTOUT)                                        2150
              GSMSEC = GSMSEC +TTOUT-TTIN                               2160
215     C                                                               2170
        C  REVISE THE CURRENT LINK FLOW                                 2180
              DO 203 I= 1,NLINK                                         2190
              VCUM(I)=VCUM(I) + LAMBDA*FLOW(I)                          2200
              TRAV(I)=USERCT(I)                                         2210
220       203 CONTINUE                                                  2220
        C                                                               2230
              IF(BOUND.LT.OBF+SUM) BOUND=OBF+SUM                        2240
              GAP=100.*DABS(OBF-BOUND)/OBF                              2250
              OBF=VAL+CONST                                             2260
225           CALL SECOND(TTOUT)                                        2270
              TTOUT=TTOUT-TBEGIN                                        2280
              WRITE(6,3000) NUMCYC,OBF,GAP,LAMBDA,TTOUT                 2290
        C                                                               2300
        C      STOPPING CRITERION CHECKING                              2310
230           IF(DABS(SUM).GT.EPSI)GOTO 210                             2320
              WRITE(6,3004) EPSI                                        2330
              GOTO 300                                                  2340
        C                                                               2350
        C CHECK THE NUMBER OF ALLOWABLE CYCLES                          2360
235       210 IF(NUMCYC.LT.MAXCYC)GOTO 220                              2370
              WRITE(6,1003) MAXCYC                                      2380
              GOTO 300                                                  2390
        C                                                               2400
        C CHECK THE RUNNING TIME                                        2410
240       220 NUMCYC=NUMCYC + 1                                         2420
              IF(TTOUT.LT.TIMEMA)GOTO 200                               2430
              WRITE(6,1002) TIMEMA                                      2440
        C                                                               2450
        C                                                               2460
```

```
      300 CONTINUE                                                              2470
          WRITE(6,2003) TITLE                                                   2480
          WRITE(6,1013) EPSI,ACCUR,TIMEMA,MAXCYC                                2490
          WRITE(6,1019) (RUNSEC(I),I=1,MXRUNS)                                  2500
      C                                                                         2510
      C--------CALCULATE TOTAL VEHICLE HOURS AND MILES                          2520
          VEHRS = 0.0                                                           2530
          VEMLS = 0.0                                                           2540
          DO 360 I=1,NLINK                                                      2550
      C                                                                         2560
      C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  2570
      C         SUPPLIED BY USER                                                2580
      C                                                                         2590
      C         ** SAMPLE PROBLEM CODE **                                       2600
      C                                                                         2610
      C--------IGNORE DUMMY LINKS (DELAY FUNCTION 1)                           2620
          IF (MOD(NULACO(I),1000) .EQ.1) GOTO 360                               2630
      C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  2640
      C                                                                         2650
      C                                                                         2660
        352 VEHRS = VEHRS + VCUM(I) * TRAV(I)                                   2670
          VEMLS = VEMLS + VCUM(I) * DIST(I)                                     2680
        360 CONTINUE                                                            2690
          VEHRS = VEHRS / 60.                                                   2700
          VEMLS = VEMLS / 100.                                                  2710
          VESPD = VEMLS / VEHRS                                                 2720
      C                                                                         2730
          WRITE(6,1014) VEHRS,VEMLS,VESPD                                       2740
      C                                                                         2750
          IF(IPRT.LE.0)GOTO 100                                                 2760
      C                                                                         2770
      C------PRINT OUT THE CALCULATED LINK FLOWS AND TRAVEL TIMES               2780
      C                                                                         2790
      C                                                                         2800
        317 REWIND NETFIL                                                       2810
          REWIND SOLFIL                                                         2820
          READ (NETFIL,4001) (LLINE(I),I=1,40)                                  2830
          WRITE(SOLFIL,4001) (LLINE(I),I=1,40)                                  2840
          NLINE = 45                                                            2850
          NPAGE = 1                                                             2860
          DO 90 N = 1,NLINK                                                     2870
      C--------PRINT PAGE HEADING EVERY 45 LINES                                2880
          IF (NLINE .GE.45) GOTO 80                                             2890
          GOTO 82                                                               2900
         80     WRITE(6,2000) NPAGE,(TITLE(JJ),JJ=1,50)                         2910
                NLINE = 0                                                       2920
                NPAGE = NPAGE + 1                                               2930
      C                                                                         2940
         82 NLINE = NLINE + 1                                                   2950
          READ (NETFIL,2004) LINE                                               2960
          ADIST = FLOAT(IDIST) / 100.0                                          2970
          ANULA=FLOAT(NULA)/10.0                                                2980
          SPEED = FLOAT(ISPEED) / 10.0                                          2990
          TOBS = FLOAT(ITOBS) / 100.0                                           3000
          TPRED = TRAV(N)                                                       3010
          IVPRED = IFIX(VCUM(N))                                                3020
          IF (IVOBS .GT. 0) GO TO 85                                            3030
              WRITE(6,2006) N,NORIG,NDEST,ADIST,ANULA,DEFUNC,SPEED,TOBS,        3040
         +                  LTYPE,TPRED,IVPRED                                 3050
              GO TO 90                                                          3060
        85 WRITE(6,2005) N,NORIG,NDEST,ADIST,ANULA,DEFUNC,SPEED,TOBS,           3070
```

```
     +                IVOBS,LTYPE,TPRED,IVPRED                                3080
      90 CONTINUE                                                             3090
         WRITE(6,2014)                                                        3100
         DO 95 I=1,NDATA                                                      3110
310      95 WRITE(6,2015)I,ALPHA(I),BETA(I),GAMMA(I),DELTA(I)                 3120
        100 CONTINUE                                                          3130
         WRITE(SOLFIL,2011)(VCUM(I),I=1,NLINK)                                3140
         WRITE(SOLFIL,2012) (TRAV(I),I=1,NLINK)                               3150
         WRITE(SOLFIL,2013) (KD(I),I=1,NCENT)                                 3160
315      REWIND SOLFIL                                                        3170
         STOP                                                                 3180
         END                                                                  3190
```

```
      SUBROUTINE  MINTRE(NN)                                                    3200
C                                                                               3210
C--------CONSTRUCT SHORTEST PATH TO ALL DESTINATIONS FROM ORIGIN NN.           3220
C                                                                               3230
C     LPR(N)  WILL CONTAIN THE NUMBER OF THE LINK INCIDENT                      3240
C             TO NODE I, BELONGING TO A SHORTEST PATH ORIGINATING               3250
C             AT ORIGIN NN (PERMANENT LABEL OF NODE).                           3260
C     LAB(K)  WILL CONTAIN THE CURRENT ADDRESS OF A NODE K IF                   3270
C             IT IS IN THE HEAP OF CANDIDATES SPECIFIED IN NSET, TSET.          3280
C     NSET AND TSET CONTAIN THE VALUES ASSOCIATED WITH NODE (K)                 3290
C             IN BINARY HEAP ORDER>                                             3300
C     NSET(I) CONTAINS THE NUMBER OF THE LINK INCIDENT TO NODE (K)>             3310
C     TSET CONTAINS THE PATH TIME TO THE NODE (K)>                              3320
C                                                                               3330
      COMMON /NET1/ NNODE,NLINK,NCENT,NACTIV,KD(150),NOLAB(1300),               3340
     +              NO(3000),ND(3000),NULACO(3000),DIST(3000)                   3350
      COMMON /NET2/ VCUM(3000),TRAV(3000), FLOW(3000),ODIJ(150)                 3360
      COMMON        WKMEM(4300)                                                 3370
      COMMON /PARAM/ EPSI , TIMEMA, MAXCYC, MAXIT, IPCH, IPRT,IPOST            3380
      COMMON /STAT/ RUNSEC(10), ITERAT(10)                                      3390
      COMMON/COEF/  ALPHA(20),BETA(20),GAMMA(20),DELTA(20),NDATA                3400
      DIMENSION LPR(1300),TSUM(1300),LAB(1300),TSET(200),NSET(200)              3410
      EQUIVALENCE(TSUM(1),WKMEM(1)),(LAB(1),WKMEM(1301))                        3420
      EQUIVALENCE(TSET(1),WKMEM(2801)),(NSET(1),WKMEM(2601))                    3430
      EQUIVALENCE(LPR(1),WKMEM(3001))                                           3440
      EQUIVALENCE (RDSEC,RUNSEC(1)),(TSLSEC,RUNSEC(2)),                         3450
     +            (PRTSEC,RUNSEC(3)),(GSMSEC,RUNSEC(4)),                       3460
     +            (PTHSEC,RUNSEC(5))                                           3470
  100 FORMAT(//,5X,45HE06 NSET IN MINTRE OVERFLOW,CURRENT ORIGIN IS,I5)         3480
C                                                                               3490
C      MIN PATH USING DIJKSTRA ALGORITHM AND HEAPS STRUCTURE                    3500
      CALL SECOND(TTIN)                                                         3510
C                                                                               3520
C--------INITIALISATION                                                        3530
C                                                                               3540
      N=NN                                                                      3550
      NA=1                                                                      3560
      NB=0                                                                      3570
C                                                                               3580
C     INITIALISE TSUM AND LAB                                                   3590
      DO 2 I=1,NNODE                                                            3600
      TSUM(I)=1.0E10                                                            3610
      LAB(I)=0                                                                  3620
      LPR(I)=0                                                                  3630
    2 CONTINUE                                                                  3640
C                                                                               3650
      TSUM(N)=0                                                                 3660
      LPR(N)=999999                                                             3670
C                                                                               3680
C                                                                               3690
C                                                                               3700
C    LABEL NEW NODES                                                            3710
C                                                                               3720
    6 J=NOLAB(N)                                                                3730
C                                                                               3740
      IF(J.EQ.0)GOTO 13                                                         3750
C                                                                               3760
```

```
                DO 7 I=J,NLINK                                          3770
                IF(NO(I).EQ.N)GOTO 8                                    3780
60              GOTO 13                                                 3790
              8 K=ND(I)                                                 3800
                IF(TSUM(K).GT.0.9E10)GOTO 9                             3810
                GOTO 7                                                  3820
              9 I13=MOD(NULACO(I),1000)                                 3830
65              IF(I13.EQ.13)GOTO 31                                    3840
                GOTO 32                                                 3850
             31 LASTI=LPR(N)                                            3860
                I13=MOD(NULACO(LASTI),1000)                             3870
                IF(I13.NE.13)GOTO 32                                    3880
70              VAL=TSUM(N)+1000.0                                      3890
                GOTO 332                                                3900
             32  VAL=TSUM(N)+TRAV(I)                                    3910
            332 IF(LAB(K).EQ.0)GOTO 10                                  3920
                GOTO 11                                                 3930
75           10 NB=NB+1                                                 3940
                LR=NB                                                   3950
                IF(NB.LE.200)GOTO 33                                    3960
                WRITE(6,100) NN                                         3970
                CALL EXIT                                               3980
80           33 LAB(K)=NB                                               3990
                GOTO 12                                                 4000
             11 LR=LAB(K)                                               4010
                IF(VAL.GE.TSET(LR))GOTO 7                               4020
             12 TSET(LR)=VAL                                            4030
85              NSET(LR)=I                                              4040
                JJ=LR                                                   4050
          C                                                             4060
          C   SIFT UP                                                   4070
          C                                                             4080
90           20 II=JJ                                                   4090
                IF(II.EQ.1)GOTO 22                                      4100
                JJ=II/2                                                 4110
                IF(VAL.LT.TSET(JJ))GOTO 21                              4120
                GOTO 22                                                 4130
95           21 TSET(II)=TSET(JJ)                                       4140
                NSET(II)=NSET(JJ)                                       4150
                KK=NSET(II)                                             4160
                KK=ND(KK)                                               4170
                LAB(KK)=II                                              4180
100             GOTO 20                                                 4190
             22 TSET(II)=VAL                                            4200
                NSET(II)=I                                              4210
                LAB(K)=II                                               4220
              7 CONTINUE                                                4230
105       C                                                             4240
          C                                                             4250
          C                                                             4260
          C   PERMANENT LABEL                                          4270
          C                                                             4280
110          13 CONTINUE                                                4290
                IF(NB.LE.0)GOTO 40                                      4300
             42 CONTINUE                                                4310
                I=NSET(1)                                               4320
                N=ND(I)                                                 4330
```

```
      TSUM(N)=TSET(1)                                                    4340
      LPR(N)=I                                                           4350
      NA=NA+1                                                            4360
      IF(NA.GE.NNODE)GOTO 40                                             4370
C                                                                        4380
C                                                                        4390
C     DELETE THE LAST NODE LABELLED                                      4400
C                                                                        4410
      TSET(1)=TSET(NB)                                                   4420
      VAL=TSET(1)                                                        4430
      NSET(1)=NSET(NB)                                                   4440
      I=NSET(1)                                                          4450
      NB=NB-1                                                            4460
      K=ND(I)                                                            4470
      LAB(K)=1                                                           4480
      JJ=1                                                               4490
C                                                                        4500
C                                                                        4510
C    PUSH DOWN                                                           4520
C                                                                        4530
   14 II=JJ                                                              4540
      JJ=II*2                                                            4550
      IF(JJ-NB)16,17,18                                                  4560
   16 IF(TSET(JJ).GT.TSET(JJ+1))JJ=JJ+1                                  4570
   17 IF(VAL.LE.TSET(JJ))GOTO 18                                         4580
      TSET(II)=TSET(JJ)                                                  4590
      NSET(II)=NSET(JJ)                                                  4600
      KK=NSET(II)                                                        4610
      KK=ND(KK)                                                          4620
      LAB(KK)=II                                                         4630
      GOTO 14                                                            4640
   18 TSET(II)=VAL                                                       4650
      NSET(II)=I                                                         4660
      LAB(K)=II                                                          4670
C     NO PATH IS ALLLOWED TO PASS THROUGH A CENTROID                     4680
C      (ORIGIN OR DESTINATION)                                           4690
      IF(N.LE.NCENT)GOTO 13                                              4700
      GOTO 6                                                             4710
   40 CALL SECOND(TTOUT)                                                 4720
      PTHSEC=PTHSEC+TTOUT-TTIN                                           4730
      RETURN                                                             4740
      END                                                                4750
```

```
      DOUBLE PRECISION FUNCTION SUMTRV(LAMBDA)                                  4760
C                                                                               4770
C--------CALCULATE OBJECTIVE FUNCTION (SUM TRAVEL TIME) FOR A                   4780
C          GIVEN LAMBDA.   (FUNCTION Z IN ALGORITHM STEP 4)                     4790
C                                                                               4800
      COMMON /NET1/ NNODE,NLINK,NCENT,NACTIV,KD(150),NOLAB(1300),               4810
     +              NO(3000),ND(3000),NULACO(3000),DIST(3000)                   4820
      COMMON /NET2/ VCUM(3000),TRAV(3000), FLOW(3000),ODIJ(150)                 4830
      COMMON        WKMEM(4300)                                                 4840
      COMMON /PARAM/ EPSI , TIMEMA, MAXCYC, MAXIT, IPCH, IPRT,IPOST            4850
      COMMON /STAT/ RUNSEC(10), ITERAT(10)                                      4860
      COMMON/COEF/  ALPHA(20),BETA(20),GAMMA(20),DELTA(20),NDATA                4870
      DOUBLE PRECISION LAMBDA,SUM,VOL                                           4880
C                                                                               4890
C                                                                               4900
      SUM=0.0                                                                   4910
      DO 100 J=1,NLINK                                                          4920
      VOL=VCUM(J) + LAMBDA*FLOW(J)                                              4930
C--------UNPACK DELAY CURVE NUMBER  JC                                          4940
      JC=MOD(NULACO(J),1000)                                                    4950
      NULA=NULACO(J)/1000                                                       4960
      CALL FINDCC(JC,SPDFF,CAP,A,B)                                             4970
C                                                                               4980
C                                                                               4990
C                                                                               5000
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  5010
C                                                                               5020
C     SUPPLIED BY USER                                                          5030
C                                                                               5040
C                                                                               5050
C                                                                               5060
C     ** SAMPLE PROBLEM CODE **                                                 5070
C                                                                               5080
C                                                                               5090
      IF(JC.EQ.1) GOTO 100                                                      5100
      IF(SPDFF.EQ.0.0) GOTO 100                                                 5110
      IF(A.EQ.0.0) GOTO 20                                                      5120
      CAP=CAP*FLOAT(NULA)/10.                                                   5130
      IF(CAP.EQ.0.0) GOTO 100                                                   5140
      IF(VOL.LT.0.) VOL=0.0                                                     5150
      SUM=SUM+(DIST(J)*0.6/SPDFF*VOL*(1.0+A*((VOL/CAP)**B)/(B+1.0)))            5160
      GOTO 100                                                                  5170
   20 SUM=SUM+(DIST(J)*0.6/SPDFF*VOL)                                           5180
C                                                                               5190
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  5200
C                                                                               5210
  100 CONTINUE                                                                  5220
      SUMTRV=SUM                                                                5230
      RETURN                                                                    5240
      END                                                                       5250
```

```
      REAL FUNCTION USERCT( J )                                                 5260
C                                                                               5270
C--------CALCULATE USER COST = TRAVEL TIME FOR LINK J                           5280
C                                                                               5290
      COMMON /NET1/ NNODE,NLINK,NCENT,NACTIV,KD(150),NOLAB(1300),               5300
     +              NO(3000),ND(3000),NULACO(3000),DIST(3000)                   5310
      COMMON /NET2/ VCUM(3000),TRAV(3000), FLOW(3000),ODIJ(150)                 5320
      COMMON /COEF/ALPHA(20),BETA(20),GAMMA(20),DELTA(20),NDATA                 5330
C                                                                               5340
      VOL=VCUM(J)                                                               5350
C       UNPACK DELAY CURVE INDEX JC                                             5360
      JC=MOD(NULACO(J),1000)                                                    5370
      NULA=NULACO(J)/1000                                                       5380
      CALL FINDCC(JC,SPDFF,CAP,A,B)                                             5390
C                                                                               5400
C                                                                               5410
C                                                                               5420
C                                                                               5430
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  5440
C     SUPPLIED BY USER                                                          5450
C                                                                               5460
C                                                                               5470
C     ** SAMPLE PROBLEM CODE***                                                 5480
C                                                                               5490
C                                                                               5500
      IF(JC.EQ.1) GOTO 100                                                      5510
      IF(SPDFF.EQ.0.0) GOTO 100                                                 5520
      IF(A.EQ.0.0) GOTO 20                                                      5530
      CAP=CAP*FLOAT(NULA)/10.                                                   5540
      IF(CAP.EQ.0.0) GOTO 100                                                   5550
      IF(VOL.LT.0.) VOL=0.0                                                     5560
C                                                                               5570
C     BPR DELAY CURVE                                                           5580
C                                                                               5590
      USERCT=DIST(J)*0.6/SPDFF*(1.0+A*(VOL/CAP)**B)                             5600
      RETURN                                                                    5610
   20 USERCT=DIST(J)*0.6/SPDFF                                                  5620
      RETURN                                                                    5630
  100 USERCT=0.0                                                                5640
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  5650
      RETURN                                                                    5660
      END                                                                       5670
                                                                                5680
```

```
      SUBROUTINE FINDCC(LTYPE,SPDFF,CAP,A,B)                                5690
      COMMON /COEF/  ALPHA(20),BETA(20),GAMMA(20),DELTA(20),NDATA           5700
      LT=LTYPE                                                              5710
C                                                                           5720
C                                                                           5730
C                                                                           5740
C                                                                           5750
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++5760
C     SUPPLIED BY USER                                                      5770
C                                                                           5780
C                                                                           5790
C     ** SAMPLE PROBLEM CODE***                                             5800
C                                                                           5810
C                                                                           5820
      SPDFF=GAMMA(LT)                                                       5830
      CAP=DELTA(LT)                                                         5840
      A=ALPHA(LT)                                                           5850
      B=BETA(LT)                                                            5860
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++5870
      RETURN                                                                5880
      END                                                                   5890
```

```
      SUBROUTINE GSM(LAMBDA,DELTA,ACCUR,VAL)                                5900
C                                                                           5910
C     - GOLDEN SECTION METHOD -  SEARCH                                     5920
C                                                                           5930
C         FIND LAMBDA IN THE INTERVAL ZERO TO "DELTA" SUCH THAT             5940
C          THE OBJECTIVE FUNCTION (SUMTRV) IS MINIMUM.                      5950
C         VAL IS SET TO THE CORRESPONDING OBJECTIVE FUNCTION                5960
C              VALUE.                                                       5970
C         ACCUR IS THE REQUIRED ACCURACY, THE SEARCH TERMINATES             5980
C               WHEN THE INTERVAL BETWEEN THE LOWER AND UPPER BOUNDS        5990
C               OF LAMBDA IS WITHIN THE REQUIRED ACCURACY.                  6000
C                                                                           6010
      DOUBLE PRECISION LAMBDA,TAU,G,D,VAL,XB,XA,FXA,FXB,SUMTRV,ACCUR        6020
C                                                                           6030
      TAU=1.6180339887499                                                   6040
      G=0.0                                                                 6050
      D=DELTA                                                               6060
      XB=D/TAU                                                              6070
      XA=D-XB                                                               6080
      FXA=SUMTRV(XA)                                                        6090
      FXB=SUMTRV(XB)                                                        6100
   77 IF(FXA-FXB)80,80,78                                                   6110
   78    G=XA                                                             6120
         XA=XB                                                            6130
         XB=G+D-XA                                                        6140
         FXA=FXB                                                          6150
         FXB=SUMTRV(XB)                                                   6160
         GOTO 82                                                          6170
   80 D=XB                                                                6180
      XB=XA                                                               6190
      XA=G+D-XB                                                           6200
      FXB=FXA                                                             6210
      FXA=SUMTRV(XA)                                                      6220
   82 IF(D-G-ACCUR)83,83,77                                               6230
C                                                                         6240
   83 LAMBDA=XB                                                           6250
      VAL=FXB                                                             6260
      IF(FXB.LT.FXA)RETURN                                                6270
      VAL=FXA                                                             6280
      LAMBDA=XA                                                           6290
      RETURN                                                              6300
      END                                                                 6310
```

```
      SUBROUTINE NETWRK(ISOL)                                                  6320
C                                                                              6330
C--------READ IN NETWORK DATA FOR ALL LINKS                                    6340
C                                                                              6350
      COMMON /NET1/ NNODE,NLINK,NCENT,NACTIV,KD(150),NOLAB(1300),              6360
     +              NO(3000),ND(3000),NULACO(3000),DIST(3000)                   6370
      COMMON /NET2/ VCUM(3000),TRAV(3000), FLOW(3000),ODIJ(150)                6380
      COMMON        WKMEM(4300)                                                6390
      COMMON /PARAM/ EPSI , TIMEMA, MAXCYC, MAXIT, IPCH, IPRT,IPOST            6400
      COMMON /STAT/ RUNSEC(10), ITERAT(10)                                     6410
      COMMON /COEF/  ALPHA(20),BETA(20),GAMMA(20),DELTA(20),NDATA              6420
      COMMON /EDIT/LINE(11),L1(26),L2(26),TITLE(50)                            6430
      COMMON /UNIT/ CARD,ODFIL,NETFIL,SOLFIL                                   6440
      INTEGER CARD,ODFIL,NETFIL,SOLFIL                                         6450
      DIMENSION JDEST(150),JDEM(150)                                           6460
      DIMENSION LLINE(40)                                                      6470
      INTEGER DEFUNC                                                           6480
      EQUIVALENCE (NORIG,LINE(1)),(NDEST,LINE(2)),(IDIST,LINE(3)),             6490
     +            (NULA,LINE(4)),(DEFUNC,LINE(5))                            6500
C                                                                              6510
      DATA MAXERR/3/                                                           6520
      DATA IHTI,IHT1,IHB,IHT2/2HTI,2HT1,1H ,2HT2/                              6530
      DATA IHPA,IHRM,IHSI,IHZE/2HPA,2HRM,2HSI,2HZE/                            6540
C                                                                              6550
 1000 FORMAT(40A2)                                                             6560
 1001 FORMAT(2A2,A1,F10.0,F5.0,10I5)                                           6570
 1002 FORMAT(2A2,A1,15I5)                                                      6580
 1003 FORMAT(5X,3(I4,1X),I3,I2,1X,I4,1X,2(I5,1X),3(I3,1X))                     6590
 1010 FORMAT(2A2,A1,20A2,/,2A2,A1,30A2)                                        6600
 2001 FORMAT(//,5X,28HE03 LINKS NOT SORTED. LINE :,I4)                         6610
 2002 FORMAT(/5X,26HE04 NO. CARDS .NE. NLINKS ,2I8)                            6620
 2051 FORMAT (5X,26HE11 TITLE CARDS MISSING : ,2A2,A1,5X,2A2,2A2,A1)           6630
 2052 FORMAT (5X,29HE12 PARAMETER CARD MISSING : ,2A2,A1)                      6640
 2053 FORMAT (5X,24HE13 SIZE CARD MISSING : ,2A2,A1)                           6650
 2060 FORMAT (5X,27HE20 TOO MANY INPUT ERRORS :,I5)                            6660
 3001 FORMAT(2I5,10(I3,I4))                                                    6670
 3011 FORMAT(/,10X,35HE05 ERROR IN INPUT OD MATRIX. TYPE:,I3,                  6680
     +       8H  CARD :,I5,10H  ORIGIN :,I5)                                   6690
 3500 FORMAT(5X,39HE21 ERROR IN DELAY FUNCTION DATA, TYPE:,I3)                 6700
 3501 FORMAT(5X,50HE22 DELAY FUNCTION COEFFICIENTS NOT SORTED. LINE: ,         6710
     +           I3)                                                           6720
 3502 FORMAT(I5,4F10.5)                                                        6730
 4001 FORMAT(/10X,26HI07 PARTIAL SOLUTION INPUT,I3)                            6740
 4004 FORMAT(10F8.0)                                                           6750
C                                                                              6760
      IF(ISOL.NE.0) GOTO 200                                                   6770
      MAXFT=0                                                                  6780
      ERR=0                                                                    6790
      ICARD=0                                                                  6800
      READ(CARD,1010) IDD1,IDD2,IDD3,(TITLE(I),I=1,20),ID21,ID22,ID23,         6810
     +               (TITLE(I),I=21,50)                                        6820
      IF(IDD1.NE.IHTI.OR.IDD2.NE.IHT1.OR.IDD3.NE.IHB.OR.ID21.NE.IHTI.OR.       6830
     +      ID22.NE.IHT2.OR.ID23.NE.IHB)GOTO 10                                6840
      GOTO 12                                                                  6850
   10     WRITE(6,2051) IDD1,IDD2,IDD3,ID21,ID22,ID23                         6860
          IERR = IERR + 1                                                      6870
   12 READ(CARD,1000) (LLINE(I),I=1,40)                                        6880
```

```
      REWIND NETFIL                                                     6890
      WRITE(NETFIL,1000) (LLINE(I),I=1,40)                              6900
60    C                                                                 6910
      C--------READ ALGORITHM CONTROL PARAMETERS                        6920
      C                                                                 6930
      READ(CARD,1001)  ID1,ID2,ID3,EPSI,TIMEMA,MAXCYC,MAXIT,IPOST,IPCH, 6940
     +                                IPRT                            6950
65    IF(ID1.NE.IHPA.OR.ID2.NE.IHRM.OR.ID3.NE.IHB)GOTO 20               6960
      GOTO 22                                                           6970
   20     WRITE(6,2052) ID1,ID2,ID3                                    6980
          IERR = IERR + 1                                              6990
   22 IF(EPSI.LE.0.0) EPSI=0.05                                        7000
70    IF(TIMEMA.LE.0.0) TIMEMA=150.0                                   7010
      IF(MAXCYC.LE.0) MAXCYC=10                                        7020
      IF(MAXIT.LE.0) MAXIT=15                                          7030
      C                                                                7040
      C--------READ NETWORK DATA AND CREATE NETWORK FILE               7050
75    READ(CARD,1002)  ID1,ID2,ID3,NNODE,NLINK,NCENT,NDATA             7060
      IF(ID1.NE.IHSI.OR.ID2.NE.IHZE.OR.ID3.NE.IHB)GOTO 30              7070
      GOTO 32                                                          7080
   30     WRITE(6,2053) ID1,ID2,ID3                                    7090
          IERR = IERR + 1                                              7100
80    32 IF (IERR .GE. MAXERR) GOTO 34                                 7110
      GOTO 36                                                          7120
   34     WRITE(6,2060) IERR                                           7130
          CALL EXIT                                                    7140
      C                                                                7150
85    36 DO 50 I=1,NNODE                                               7160
   50     NOLAB(I)=0                                                   7170
      NOLAB(1)=1                                                       7180
      I=0                                                              7190
      J=1                                                              7200
90    55 I=I+1                                                         7210
      READ(CARD,1003) LINE                                             7220
      C                                                                7230
      C     TEST BLANK LINE AS                                         7240
      C       END OF NETWORK DATA                                      7250
95    C                                                                7260
      IF(NORIG.EQ.0.AND.NDEST.EQ.0) GOTO 59                            7270
   56 NO(I) = NORIG                                                    7280
      ND(I) = NDEST                                                    7290
      DIST(I)=IDIST                                                    7300
100   IF(MAXFT.LT.DEFUNC) MAXFT=DEFUNC                                 7310
      NULACO(I)=NULA*1000+DEFUNC                                       7320
      WRITE(8,1003) LINE                                               7330
      C--------SET UP LINK ORIGIN INDEX TABLE NOLAB                     7340
      IF(NO(I)-J)57,55,58                                              7350
105      57     WRITE(6,2001) I                                       7360
             CALL EXIT                                                 7370
      58 J=NO(I)                                                       7380
      NOLAB(J)=I                                                       7390
      GOTO 55                                                          7400
110   C                                                                7410
      59 I=I-1                                                         7420
      IF(I.NE.NLINK)GOTO 60                                            7430
      GOTO 61                                                          7440
   60     WRITE(6,2002) I,NLINK                                        7450
```

```
                CALL EXIT                                                       7460
             61 CONTINUE                                                        7470
                NACTIV=0                                                        7480
                NOLD=0                                                          7490
                DO 102 IO=1,NCENT                                               7500
      C                                                                         7510
      C       READ IN THE DEMANDS ASSOCIATED WITH THE CURRENT ORIGIN            7520
      C                                                                         7530
      C       KD WILL CONTAIN ORIGIN NODE NUMBER OF EACH ORIGIN IN THE          7540
      C          OD MATRIX IN THE ORDER READ IN. KD ENTRY WILL BE NEGATIVE      7550
      C          IF THERE IS NO DEMAND FROM THAT ORIGIN.                        7560
      C                                                                         7570
                DO 101 I=1,NCENT                                                7580
            101    ODIJ(I)=0.00                                                 7590
                M=10                                                            7600
                ICARD=ICARD+1                                                   7610
                READ(CARD,3001) N,NT,(L1(J),L2(J),J=1,10)                       7620
      C                                                                         7630
      C       TEST BLANK LINE AS                                                7640
      C         END OF MATRIX DATA                                              7650
      C                                                                         7660
                IF(N.EQ.0.AND.NT.EQ.0) GOTO 105                                 7670
                GOTO 106                                                        7680
            105    IERR = 2                                                     7690
                   GO TO 113                                                    7700
            106 IF(N.EQ.0.OR.N.EQ.NOLD)GOTO 111                                 7710
                GOTO 112                                                        7720
            111    IERR=1                                                       7730
            113    WRITE(6,3011) IERR,ICARD,IO                                  7740
                   CALL EXIT                                                    7750
            112 KD(IO)=-N                                                       7760
                NOLD=N                                                          7770
      C--------IF NO DEMAND FOR THIS ORIGIN KD ENTRY NEGATIVE                   7780
                IF(NT.EQ.0)GOTO 102                                             7790
                KD(IO)=N                                                        7800
      C--------SET UP DEMAND MATRIX ODIJ FOR THIS ORIGIN                        7810
                DO 120 I=1,10                                                   7820
                JDEST(I)=L1(I)                                                  7830
            120 JDEM(I)=L2(I)                                                   7840
                IF(NT.LE.10)GOTO 150                                            7850
                NUM=NT/10                                                       7860
                IF(NUM*10.EQ.NT) NUM=NUM-1                                      7870
                DO 121 I=1,NUM                                                  7880
                ICARD=ICARD+1                                                   7890
                READ(CARD,3001) N,NT,(L1(J),L2(J),J=1,10)                       7900
      C                                                                         7910
      C       TEST BLANK LINE AS                                                7920
      C         END OF MATRIX DATA                                              7930
      C                                                                         7940
                IF(N.EQ.0.AND.NT.EQ.0) GOTO 105                                 7950
            126 IF(N.NE.NOLD)GOTO 111                                           7960
                DO 122 J=1,10                                                   7970
                M=M+1                                                           7980
                JDEST(M)=L1(J)                                                  7990
            122 JDEM(M)=L2(J)                                                   8000
            121 CONTINUE                                                        8010
      C                                                                         8020
```

```
      C                                                                 8030
        150  CONTINUE                                                   8040
             DO 116 J=1,NT                                              8050
175          IF(JDEST(J).EQ.N)GOTO 116                                  8060
             DEM=JDEM(J)                                                8070
             NEXT=JDEST(J)                                              8080
             ODIJ(NEXT)=DEM                                             8090
             NACTIV=NACTIV+1                                            8100
180     116  CONTINUE                                                   8110
             WRITE(ODFIL) ODIJ                                          8120
        102  CONTINUE                                                   8130
      C                                                                 8140
      C      DELAY FUNCTION DATA                                        8150
185   C                                                                 8160
             IF(NDATA.GT.0) GOTO 170                                    8170
             IERR=1                                                     8180
        165      WRITE(6,3500) IERR                                     8190
             CALL EXIT                                                  8200
190     170  IF(NDATA.GE.MAXFT) GOTO 175                               8210
             IERR=2                                                     8220
             GOTO 165                                                   8230
        175  DO 180 I=1,NDATA                                          8240
             READ(CARD,3502) J,XAL,XBE,XGA,XDE                          8250
195          IF(J.EQ.I) GOTO 178                                        8260
             WRITE(6,3501) I                                            8270
             CALL EXIT                                                  8280
        178  ALPHA(J)=XAL                                               8290
             BETA(J)=XBE                                                8300
200          GAMMA(J)=XGA                                               8310
             DELTA(J)=XDE                                               8320
        180  CONTINUE                                                   8330
      C                                                                 8340
             RETURN                                                     8350
205   C                                                                 8360
      C                                                                 8370
      C       RE-ESTABLISH PARTIAL SOLUTION                             8380
      C                                                                 8390
      C      ENTRY PARTSOL                                              8400
210     200  CONTINUE                                                   8410
             WRITE(6,4001) IPOST                                        8420
             REWIND SOLFIL                                              8430
             READ(SOLFIL,1000) (LLINE(I),I=1,40)                        8440
             WRITE(6,1000) (LLINE(I),I=1,40)                            8450
215          READ(SOLFIL,4004) (VCUM(I),I=1,NLINK)                      8460
             DO 206 I=1,NLINK                                           8470
             FLOW(I)=0.                                                 8480
        206  TRAV(I)=USERCT(I)                                          8490
             RETURN                                                     8500
220          END                                                        8510
```

```
      SUBROUTINE LOADTR(NN)                                                  8520
C                                                                            8530
C--------ACCUMULATE FLOW ON LINKSUSED ON ROUTE TO ALL DESTINATIONS          8540
C         FROM THE GIVEN ORIGIN (NN)                                         8550
      COMMON /NET1/ NNODE,NLINK,NCENT,NACTIV,KD(150),NOLAB(1300),            8560
     +              NO(3000),ND(3000),NULACO(3000),DIST(3000)               8570
      COMMON /NET2/ VCUM(3000),TRAV(3000), FLOW(3000),ODIJ(150)              8580
      COMMON        WKMEM(4300)                                              8590
      COMMON /PARAM/ EPSI , TIMEMA, MAXCYC, MAXIT, IPCH, IPRT,IPOST          8600
      COMMON /STAT/ RUNSEC(10), ITERAT(10)                                   8610
      COMMON/COEF/  ALPHA(20),BETA(20),GAMMA(20),DELTA(20),NDATA             8620
      DIMENSION LPR(1300),TSUM(1300),LAB(1300),TSET(200),NSET(200)           8630
      EQUIVALENCE(TSUM(1),WKMEM(1)),(LAB(1),WKMEM(1301))                     8640
      EQUIVALENCE(TSET(1),WKMEM(2801)),(NSET(1),WKMEM(2601))                 8650
      EQUIVALENCE(LPR(1),WKMEM(3001))                                        8660
      INTEGER WKMEM                                                          8670
C                                                                            8680
C--------CONSTRUCT SHORTEST PATHS FROM ORIGIN NN TO ALL DESTINATIONS.        8690
      CALL MINTRE(NN)                                                        8700
      LPR(NN)=0                                                              8710
C--------TRACE BACK LINKS ON SHORTEST PATH FOR EACH DESTINATION AND           8720
C         ADD OD VOLUME TO LINK TOTAL.                                       8730
      DO 100 I=1,NCENT                                                       8740
      IF(ODIJ(I).LE.0.0)GOTO 100                                             8750
      NEXT=I                                                                 8760
  104 L=LPR(NEXT)                                                            8770
      IF(L.EQ.0)GOTO 100                                                     8780
      FLOW(L)=FLOW(L)+ODIJ(I)                                                8790
      NEXT=NO(L)                                                             8800
      IF(NEXT.EQ.NN)GOTO 100                                                 8810
      GOTO 104                                                               8820
  100 CONTINUE                                                               8830
C                                                                            8840
      RETURN                                                                 8850
      END                                                                    8860
```

APPENDIX II

SAMPLE PROBLEM INPUT FILE

```
TIT1 CONGESTED ASSIGNMENT TEST
TIT2 PROBLEM GS
 SAMPLE PROBLEM FOR ALGORITHM A1 (FEB 12 1975)
PARM       .0001    20    20          0          1
SIZE     17    28     4     2
GS74      1     5     0  10 1
GS74      3    13     0  10 1
GS74      5     6   300  10 2
GS74      5     7   200  10 2
GS74      5     8   100  10 2
GS74      6     9  2300  10 2
GS74      6    10  1700  10 2
GS74      7     6   400  10 2
GS74      7    11  1200  10 2
GS74      8     7   500  10 2
GS74      8    12   600  10 2
GS74      9     2     0  10 1
GS74     10     9  2100  10 2
GS74     10    14  1000  10 2
GS74     11    10  1600  10 2
GS74     11    15   700  10 2
GS74     12    11  1500  10 2
GS74     12    16  1100  10 2
GS74     13     8  1800  10 2
GS74     13    12  1900  10 2
GS74     13    16  2200  10 2
GS74     14     9  2000  10 2
GS74     14    17   900  10 2
GS74     15    14   800  10 2
GS74     15    17  1400  10 2
GS74     16    15  2400  10 2
GS74     16    17  1300  10 2
GS74     17     4     0  10 1

    1     2  42500  22500
    2     0
    3     2  22500  42500
    4     0
    1         0.0          0.0         60.          0.0
    2        0.02          5.0         30.       1000.0
```

APPENDICE III

SAMPLE PROBLEM OUTPUT REPORT

I09 INITIAL SOLUTION-VALUE OF OBJ. FUNCTION .4573932292E+07

CYCLE NUMBER	OBJECTIVE FUNCTION	GAP %	LAMBDA	CUMULATIVE RUN TIME (SECONDS)
1	.1056740107E+07	499.9	.4265	.65
2	.9724572081E+06	99.7	.1591	.87
3	.9615039583E+06	24.7	.0933	1.06
4	.9588442485E+06	9.8	.0557	1.25
5	.9571642875E+06	7.8	.0445	1.44
6	.9559770237E+06	6.1	.0395	1.63
7	.9551718500E+06	6.0	.0263	1.81
8	.9535017166E+06	5.6	.0639	2.01
9	.9529605557E+06	5.3	.0213	2.21
10	.9523841544E+06	4.6	.0263	2.40
11	.9519683242E+06	4.0	.0213	2.59
12	.9515849685E+06	3.5	.0232	2.75
13	.9513041837E+06	3.4	.0132	2.94
14	.9509264216E+06	2.5	.0313	3.15
15	.9506836955E+06	2.5	.0132	3.34
16	.9503861027E+06	2.5	.0213	3.52
17	.9501608136E+06	2.4	.0132	3.69
18	.9500193021E+06	2.1	.0132	3.88
19	.9497211283E+06	2.1	.0213	4.07
20	.9495065485E+06	2.1	.0163	4.24

02 RUN TERMINATED ... CYCLE LIMIT : 20

PROBLEM GS CONGESTED ASSIGNMENT TEST

OBJECTIVE FUNCTION TOLERANCE	=	.1000000E-03
LAMBDA TOLERANCE	=	.1000000E-01
TIME LIMIT	=	.2000000E+02
NUMBER OF CYCLES	=	20

CPU SECS FOR - NETWORK DATA INPUT	=	.37
- SET UP INITIAL SOLUTION	=	.06
- RE-ESTABLISH PARTIAL SOLUTION	=	0.00
- GOLDEN SECTION METHOD (TOTAL)	=	3.10
- FINDING SHORTEST PATH (TOTAL)	=	.05

TOTAL VEHICLE HOURS	=	22605.88
TOTAL VEHICLE MILES	=	434068.63
AVERAGE SPEED	=	19.20

PROBLEM GS CONGESTED ASSIGNMENT TEST

LINK IDEN	NODE FROM	NODE TO	LENGTH MILES	NUMB LANES	DELAY FUNCT	OBSERVED TRAVEL SPEED (MPH)	OBSERVED TRAVEL TIME (MINS)	OBSERVED VOLUME	LINK TYPE	PREDICTED TRAVEL TIME (MINS)	PREDICTED VOLUME
1	1	5	0.00	1.0	1	0.0	0.00	NA	0	0.00	5000
2	3	13	0.00	1.0	1	0.0	0.00	NA	0	0.00	5000
3	5	6	3.00	1.0	2	0.0	0.00	NA	0	15.82	2413
4	5	7	2.00	1.0	2	0.0	0.00	NA	0	6.01	1905
5	5	8	1.00	1.0	2	0.0	0.00	NA	0	2.01	681
6	6	9	23.00	1.0	2	0.0	0.00	NA	0	110.36	2338
7	6	10	17.00	1.0	2	0.0	0.00	NA	0	48.72	1849
8	7	6	4.00	1.0	2	0.0	0.00	NA	0	10.82	1774
9	7	11	12.00	1.0	2	0.0	0.00	NA	0	28.67	1576
10	8	7	5.00	1.0	2	0.0	0.00	NA	0	11.26	1445
11	8	12	6.00	1.0	2	0.0	0.00	NA	0	12.04	681
12	9	2	0.00	1.0	1	0.0	0.00	NA	0	0.00	5000
13	10	9	21.00	1.0	2	0.0	0.00	NA	0	58.44	1812
14	10	14	10.00	1.0	2	0.0	0.00	NA	0	20.07	697
15	11	10	16.00	1.0	2	0.0	0.00	NA	0	32.08	661
16	11	15	7.00	1.0	2	0.0	0.00	NA	0	33.26	2330
17	12	11	15.00	1.0	2	0.0	0.00	NA	0	33.41	1415
18	12	16	11.00	1.0	2	0.0	0.00	NA	0	22.26	901
19	13	8	18.00	1.0	2	0.0	0.00	NA	0	40.54	1445
20	13	12	19.00	1.0	2	0.0	0.00	NA	0	46.91	1636
21	13	16	22.00	1.0	2	0.0	0.00	NA	0	66.86	1918
22	14	9	20.00	1.0	2	0.0	0.00	NA	0	40.35	848
23	14	17	9.00	1.0	2	0.0	0.00	NA	0	18.24	922
24	15	14	8.00	1.0	2	0.0	0.00	NA	0	16.46	1073
25	15	17	14.00	1.0	2	0.0	0.00	NA	0	34.39	1627
26	16	15	24.00	1.0	2	0.0	0.00	NA	0	48.01	369
27	16	17	13.00	1.0	2	0.0	0.00	NA	0	71.91	2450
28	17	4	0.00	1.0	1	0.0	0.00	NA	0	0.00	5000

DELAY FUNCTION NUMBER	DELAY FUNCTION COEFFICIENTS ALPHA	BETA	GAMMA	DELTA
1	0.0000	0.0000	60.0000	0.0000
2	.0200	5.0000	30.0000	1000.0000